Mimesis : Interactive interface for mass-interaction modeling

Matthieu Evrard, Annie Luciani, Nicolas Castagné

MIMESIS: Interactive Interface for Mass-Interaction Modeling

Matthieu Evrard
ACROE-ICA Laboratory
matthieu.evrard@imag.fr
www-acroe.imag.fr

Annie Luciani
ACROE-ICA Laboratory
annie.luciani@imag.fr
www-acroe.imag.fr

Nicolas Castagné
ACROE -ICA
nicolas.castagne@imag.fr
www-acroe.imag.fr

Abstract

While 3D graphics software and 3D shapes modeling developed considerably, there is still a need for integrated physics-based tools. This would ideally require an integrated, coherent, usable, generic, physical modeling formalism, and a dedicated software, preferably to a collection of one shot models or animation techniques. This article introduces MIMESIS, a end-user software based on mass-interaction modeling. In MIMESIS, the mass-interaction paradigm (and, more generally, animation) is the core of the creation process at hand. It joins together a comprehensible, user-friendly modeler, various simulators, various coating means for visualizing synthesized movements and a growing set of pedagogical examples and library of models.

Keywords: physically-based modeling, simulation, animation, mass-interaction network, user-friendly interface, animation language.

1 Introduction

Mass-interaction modeling has been introduced about fifteen years in pioneering researches as those of [1,2,3,4,5,6,7], proving their interest in the synthesis of a large variety of quality movements and animations.
Quite surprisingly, there is today no user-friendly integrated framework that would empower truly end-users with mass-interaction modeling, for the design of complex models and the synthesis of complex movements. In this context, Sodaplay [6], which allows assembling a mass-spring system to create moving creatures, can be seen as an exemplary toy-software. Indeed, Sodaplay focuses on specific categories of structures (the 'moving creatures'), provides a restrictive set of interactions, imposes homogeneous values for the physical parameters over the interactions, does not support complex nor big systems, does not provide tools for globally working on parts of the system.
MIMESIS software is used to design interactively mass-interaction models, facilitating the modeling of complex physically-based scene by non-physicists and animation designers. It aims at eliciting the appropriate features (core features, set of basic modules, user interface…) to empower end-users with mass-interaction modeling. On this basis, the framework joins together a comprehensible, user-friendly modeler, several real time simulators, flexible 3D coating algorithm, a growing set of pedagogical examples and a library of models.
As a first publication on MIMESIS, this article aims at offering a description of the interactive modeler in its current state, rather than discussing the theoretical and technical issues on mass-interaction method already done in lot of publications [4,8]. The first two sections give an overview of the framework, and a detailed description of the mass-interaction formalism it features. The section 4 and 5 discuss the features of the modeler, and section 6 the 3D coating processes. Finally, section 7 overviews a couple of example, illustrating some of the MIMESIS' usages.

2 An overview of MIMESIS

2.1 User process

MIMESIS lets the user operate at the most elementary level of the mass-interaction formalism. Within MIMESIS, the mass-interaction physical model is central, and comes first. The user is first proposed to think and to design the movement in terms of physical principles and abstractions. In such way of thinking and designing the motion, the visualization of the movement by defining and mapping geometrical data and shapes is the final stage in the user process. In MIMESIS, the modeling process is composed of five stages:

1/ The user builds the network of mass-interaction modules, by using an appropriate language mixing graphical and textual representations.

2/ Initial conditions (positions, velocities) of the mass elements are given.

3/ She specifies the values of the physical parameters for the physical modules.

4/ The physical model is computed within one of the simulator producing motions of set of points.

5/ The user designs the mapping of previous synthesized point motions to a 2D or 3D set of shapes by a coating process.

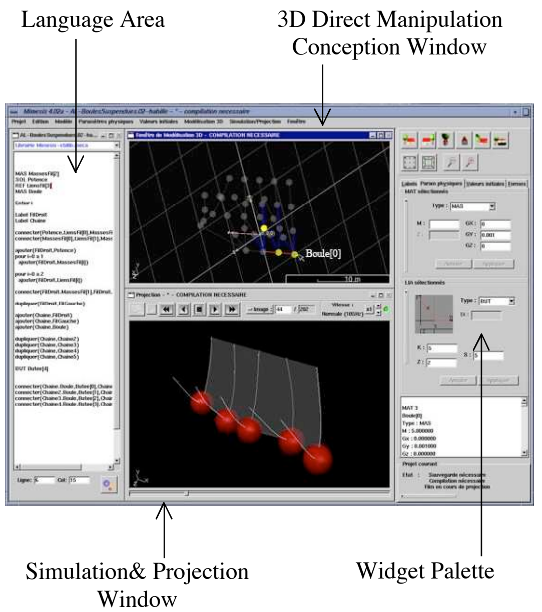

Figure 1: General view of the MIMESIS' user space

Figure 1 presents a general view of the user space, showing how each of the user stages is given a specific space and ergonomic: language area for scripting the network of modules (left), palette for manipulating through widgets the data (right), and two OpenGL-based windows for respectively working on the network in a direct-manipulation manner and visualizing the resulting movements.

Two remarks are possible when considering the MIMESIS' user process.

First, it is quite different from the user processes usually enabled by modelers/simulators in Computer Graphics, in which the modeling of 3D graphic objects comes first (except for particles systems, see §3.1), and only then the user designs various animation means (key frames, kinematics, eventually various physics based algorithms, etc.) for animating the 3D scene.

Second, while mass-springs models are usually practiced as meshes of predefined surfaces or volumes, MIMESIS points out the fact that mass-interaction modeling shifts the physically-based modeling toward network-like or mesh-free approaches.

2.2 A structural description

Figure 2 offers a structural view of the various MIMESIS' features allowing the user process to be implemented. It shows five major blocks.

Left, Top: The modeler, which is the main program in the framework. It groups the CORE that defines the core features of MIMESIS, the user language LANG for designing mass-interaction network (§4), the shape coating system (§6.1), and the graphics interface GUI (§5). The GUI incorporates the others. It allows interacting in a widget-based and direct manipulation-based manner with the model at hand. Thus, the modeler is a multimodal application, in the sense that it couples intimately a language and a GUI-based interaction for manipulating the model.

Right, Top and Bottom: Two simulators can be used when using the MIMESIS framework (§2.6): an embedded threaded simulator, and a fully synchronous, hard real-time, interactive and multisensory simulator, that runs on specific multi-processor hardware.

Centre, Bottom: various third party programs can be used from the modeler, or externally to it. Among them are various physics-based dynamic coating programs designed in the laboratory (§6.2).

Left, Bottom: Finally, a growing pedagogical set of examples and teaching courses, and a growing library of models is provided with the framework.

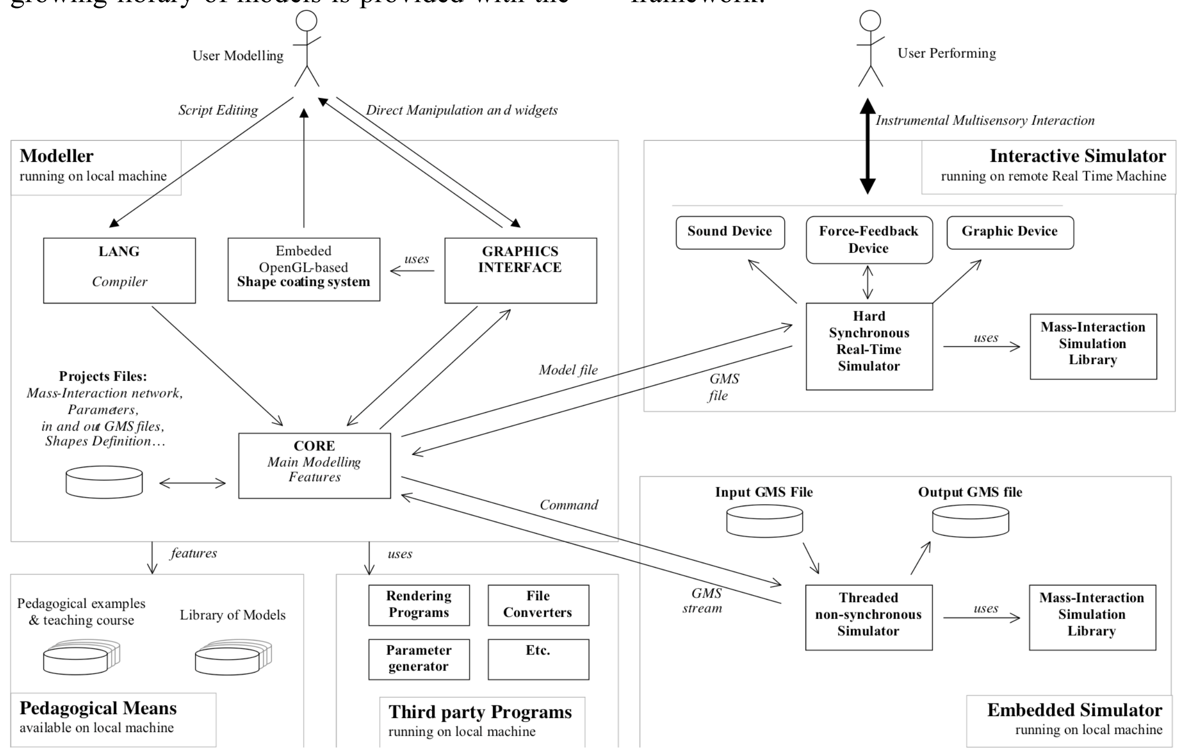

Figure 2: A structural view of MIMESIS

3 Mass-Interaction formalism and simulators

The core of MIMESIS is inspired by the mass-interaction-like formalism as proposed by Luciani & al. in [3]. It is *formalism* in the sense that it formalizes in a graceful way the concepts of masses and interactions. It is also a *language* in the sense that it defines primitive types (the mass interaction modules, which are of two sorts: MAT for material elements and LIA for liaisons, or interactions, which are connected to two MAT) and functions (connections, etc.) that allow describing fully a mass-interaction network. It is, finally, a *simulation mean*, in the sense that it comes along with an optimized algorithm for each module, and principles and software structures for computing the models.

3.1 Related works

Particle modeling, mass-interaction modeling, or the restricted mass-spring modeling, has been introduced a long time ago in Computer Graphics. Most well known approaches are:

- *Particles Systems*: non-physical particles [10], or more recently weighted but independent particles, evolve in acceleration fields. 3D animation tools today commonly feature particle systems. In this kind of systems, the coating of particle generally comes after the establishment of the systems like in MIMESIS.

- *Mass-spring meshes methods* used to animate *soft objects* [7, 9, 1]. Deformations of a deformable solid are computed by using a mass-spring mesh, usually generated from a geometrical mesh of a volume or a surface. Mass-spring meshes are usually given as an alternative to other more time consuming means, such as Finite Elements Method. They are progressively made available in 3D animation software.

The mass-interaction approach to model in MIMESIS differs from these techniques. Consisting in a specific algorithmic implementation of Newtonian mass-point physics, it relates more to works like the pioneering [11], or the more recent [12, 2, 3, 4, 5, 6].

Noticeably, it does not fall in the scope of solid physics: while modeling, the user never deals with shapes nor solids, but only with punctual masses and interactions.

The approach promotes a constructivist, network-like, mesh-free, modeling process, rather than a mesh-discretization approach. The physical modeling process starts ‘from scratch’. The user *assembles* the basic modules as a network, by handling directly the mass-interaction formalism. A geometrical mesh of a volume or a surface is not a priori given.
Along with masses, the approach features a large variety of interactions: common springs, frictions, buffers, etc., but also a set of complex interactions necessary to model various behaviors of matter, from gas to solids through fluids, gels and pastes.

3.2 The set of Physical Modules

The following offers a condensed description of the set of modules implemented in MIMESIS.
Along with the basic MAS (punctual mass), SOL (fixed point, ground), and REF (spring-friction), various non-linear interactions of various complexities are available. Interactions apply an opposite axial force on the two connected masses.
Most of these non-linear interaction modules are defined by piecewise force functions according to the elongation of the liaison, and featuring on each segment an additional viscosity parameter.
Generally speaking, each of the piecewise-based interaction modules corresponds to a more usual continuous-space interaction, such as Van Der Waals interaction, $1/r^2$ potential interaction, etc. Indeed, many past works proved that the use of piecewise approximations is truly sufficient for generating the phenomena at hand in a satisfying manner in the context of Computer Graphics, i.e. for the eye. Additionally, piecewise functions offer various advantages. For example, their algorithm is far more efficient and, on the usability side, they enable parameters, such as thresholds, elasticity on each segment, etc, that are more relevant for the user, as compared to the parameter of the corresponding algebraic formulations.
The BUT (buffer interaction) module models contact through a visco-elasticity activated when elongation falls under a specified threshold. The BUL module (bubble) models the complementary interaction, i.e. a bounding sphere.
The REP (REP3, REP4, fig 3, d and e), ATR (ATR3, ATR4, fig 3, f and g) and COH (COH3, COH4, fig 3, h and i) series model respectively more complex REPulsive ATRactive, and COHesive interactions. Past experiments proved that using a maximum of 4 segments in these piecewise-based modules offers the best balance between quality of the simulated phenomena, algorithm complexity, and usability of the module.
Anyhow, for peculiar but rare cases, the generic LLM (“piecewise linear interaction module”, fig 3 j) allows designing more complex piecewise-based interactions. A LLM groups together a non-linear viscosity and a non-linear elasticity, defined by two piecewise curves. The first curve defines the force to be applied according to the distance of the two connected MAT. On each segment, an additional viscosity parameter is available. The second curve defines the force to be applied according to the relative velocities of the two connected MAT. On each segment, an additional elasticity parameter is available. This second curve can be used, for example, to model bow-like interaction.

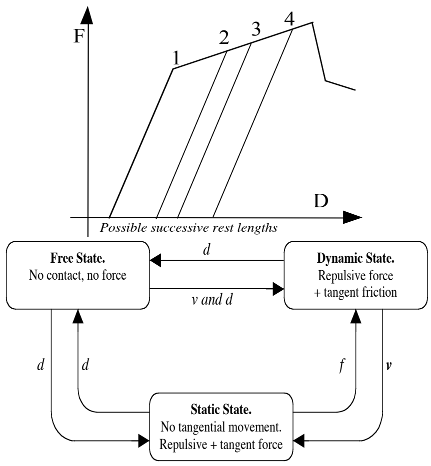

Figure 4: Up: a schematic view of the PLAST interaction. Down: finite-state automata for solid friction FROS. Transitions are made according to the variables: d: inter-mass distance; v: relative tangent speed of the two masses; f: tangent force applied

Along with the piecewise-based interactions, two other non-linear, hysteretic interactions are provided.

The PLAST module allows modeling basic plastic behavior (fig 4 up). A PLAST interaction locally behaves as a visco-elasticity, but its rest length is modified each time the computed force overpasses a given threshold.
Finally, the FROS module models simply solid friction. A FROS is based on a 3-state finite state automaton (fig 4 down). In Static and Dynamic states, conversely to all the other interactions, a FROS applies a force along the relative tangent trajectories of the two connected MAT. The norm of the applied force is proportional to the relative tangent speed of the two masses.
Noticeably, the algorithms of these two mass-to-mass (e.g. punctual) interactions are deduced from the laws proposed by Physics at a macroscopic emergent level. Though, they allow obtaining the macroscopic behaviors at hand.
The set of modules of MIMESIS has been progressively refined over many past works. It was chosen to be sufficient to cover complex behaviors of matter. It allows obtaining very diverse categories of movement, including complex emerging movement. Indeed, almost all the results that have been obtained and published for 30 years by the laboratory including models for marionettes, smokes, pasts, gel, mechanical structures, vehicles, crowds, dance movement, etc. have used nothing but the chosen modules (see examples in §7).
However, the set of modules is also minimal, in the sense that each of the module focuses without redundancy on a basic behavior of matter, and was designed to be usable. Noticeably, and though its generality, the set remains small. As a consequence, it is not too complicated for a user to experiment with each module, and internalize its specificities.
As a remark, the questions of collision detection and collision response handling, which has received much attention in Computer Graphics [13] do not apply in the context of this set of modules. In MIMESIS, collisions are inherently handled by the formalism through the non-linear optimized interactions conditioned to the relative positions of the connected masses. As an example, a potential contact between two masses would be modeled expressly with a BUT or a BUL.

3.3 **Gesture** ***as Input and Output***

A MIMESIS model is a movement generator. It produces movement as output, and eventually can handle movement as input. A specific format for Gesture and Movement Signal ™GMS has been defined for gesture stream and gesture file, and described in another publication [14]. The input GP module (Position Generator) in MIMESIS is a MAT-like element. The position of the MAT follows a specified gesture channel in the GMS input stream. The output EP module (Position Export) is a degenerated LIA module, connected to a single MAT in the model. It writes the position of the connected MAT to a channel of the output GMS stream.

3.4 MIMESIS Core simulators

Two simulators, based on “mass-interaction paradigm”, are embedded in the main modeler application of MIMESIS:
• One that is a relaxed real-time simulator, in which the relation between the gestures and actions inputs and the physically-based simulation is asynchronous. Only the dataflow is preserved. It is implemented on general-purpose platforms and it is based on a C-library of physically-based interacting modules.
• One that is a hard real-time synchronized simulator including force feedback devices. It is implemented in a dedicated multi-processor hardware.
Those simulators implement three embedded simulation loops at each basic frequency for the visual representations (basically 25 – 50 Hz), auditory behaviors (basically tens of KHz) and gestures and physical behaviors (basically about some KHz).
The simulation frequency in MIMESIS is programmable, from 0 to 44100 KHz, by multiples of the visual frequency and submultiples of the audio frequency, because the models that can be implemented may produce images, motions and sounds. In addition, the simulated models can be controlled by gestures. The default value of the simulation rate is 1050Hz. It corresponds to the average frequency bandwidth of most of physical non vibro-acoustical behaviors and it guaranties the convergence of the physical algorithms in most case. Consequently MIMESIS will guaranty the compatibility with real-time simulators integrating force feedback devices.

4 Constructing the network

4.1 Language or direct manipulation?

Modularity is a major feature of the MIMESIS approach to mass-interaction modeling. The design process is a constructive process based on a step-by-step building of a network made of tens to tens of thousand, and hundred thousands MAT and LIA modules. Depending on the user process, modules must be handled either individually or in sets, in a graceful way.
The only use of Direct Manipulation [15, 16] in this context is not sufficient depending of the type of model. Direct manipulation suits well with highly structured, heterogeneous models with small number of modules, in which user manipulates each element of the model in itself. A typical example is modeling a puppet or a vehicle. Direct manipulation is unwieldy for others types of models as fluids, large homogeneous structures (large garments, etc), wide mesh-free scenes (crowds), structures with unpredictable rearrangements (sand pilings). For these types of scenes, the use of a language for manipulating the network was far more promising, by allowing expressing in condensed manner complex or repetitive operations, manipulating/indicating easily a given structure in large sets of structures, and defining abstractions for organizing data.
Let give the example of pastes, or water and smoke, shown in figures 10 and 11: In such, all the masses are interconnected at each time by same interaction functions. Such models are impossible to design by direct point-to-point manipulation and they easy to write in the form of: MAS name, number; LIA name, type, number; Connect all to all MAS by LIA.
In addition, this way of description corresponds to the natural description of such a scene.
Consequently, these two types of manipulation are possible in MIMESIS.

4.2 Sub-Networks and Labels

Besides the obvious and quite common need of considering sub-networks (or sets of modules) in various phases of the modeling process, sub-networks handling must follow various specific requirements.
First, one should be able to add a given module to various sets, so that it can be handled later in the context of any of these sets. For instance, depending on the stage in modeling, one must be able to handle the "hub of a wheel" as part of a "wheel", but also as a part of "a whole vehicle", or as a member of "all the hubs of all vehicles".
Second, a mean for setting up a hierarchy is necessary. The user should be able to tag sets of modules must as belonging to bigger sets.
To that aim, the MIMESIS language defines a unique specific data structure called *labels*.
At first glance, a label is a name (a string) used to refer to a module, or a set of module (i.e.: a sub-network). As many labels as needed can refer to a given module, or a given set of modules. A module, or a set of module, can belong to as many sets as needed. In order to allow a hierarchical build, strings in the label namespace can be point-separated. Point-separated sub-strings depict a hierarchy.
With the label feature, the user can build up an oriented *graph of labels* as complex and flexible as needed. Labels are specified, and useful, within the script. After interpreting the script, the labels are made available in the GUI allowing a quick-selection of the contained modules.

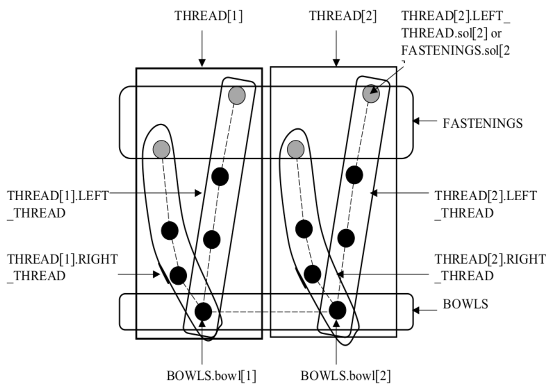

Figure 5: A simple example of the use of Labels. Labels define sets of modules. Point-separated names depict a hierarchy over le labels.

5 The MIMESIS GUI

Once the network of modules and the labels have been scripted in the MIMESIS language, and once the script has been compiled, the MIMESIS framework enters a GUI-based state. Within the GUI, along with the basic usual services (saving, canceling, etc.), the user can set up the parameters and initial conditions of the modules, control the simulations and simulation data, and design the means employed to coat the generated movement.

5.1 3D Conception Window

The GUI is organized around a "3D conception window", in which the network of modules is drawn in a diagrammatic manner in its initial state, before simulation. The conception window allows basic direct manipulations (repositioning, selecting, etc.).

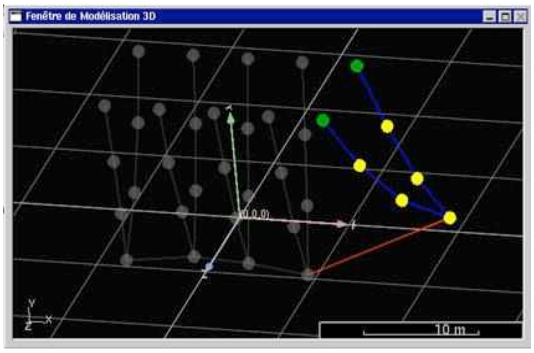

Figure 6: the diagrammatic view of a simple model in the 3D conception window.

5.2 Parameter and init manipulation

Given the huge number of modules that can be instantiated in a network, handling of physical parameters of the modules and initial condition of the masses required a special attention. Especially, tools were required to facilitate the manipulation of parameters and initial conditions of numerous modules in one shot.

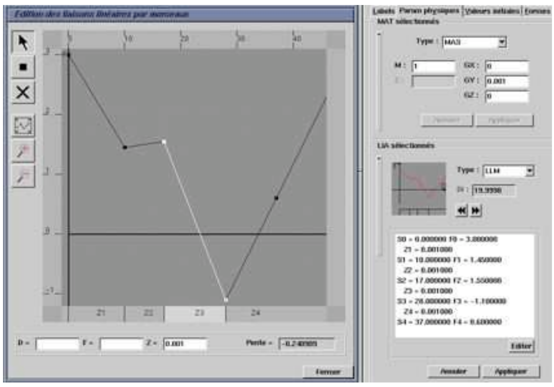

Figure 7: parameter palette (right) and graphical tool for defining LLM (left).

Three means are provided for setting up parameters and initial conditions.
The first is based on *homogeneous* conditions (fig. 7 right). In the Parameter Pallet, the user can affect a given parameter or initial conditions to all the modules of a given type in the selection.
The second is based on a full list. The Parameter Window lists the parameters of each of the selected modules, arranged by type.
The latest is made of a range of specific tools that allow acting on the parameters and initial conditions through various mathematical and geometrical relations: multiplications, random affectation, rotation & translation, etc. As an example, one of the provided tools allows computing initial conditions of a set of modules so that these are dispatched along elementary shapes: lines, squares, spheres, etc.
Finally, a graphic tool is offered for the design of the piecewise non-linear interaction functions of the LLM modules (fig. 7, left).

6 Revealing Movement through Animated Image

Once the user has defined its model, and has simulated it, the creation process by using MIMESIS enters in a new phase in which the user designs the way the movements will be revealed by using various coating means.
Basically, the goal of this stage is to reconstruct some shapes, and more generally some animated images, from the moving punctual masses. This is typically a mapping stage from evolution functions to geometrical primitives. The number of possible mapping is infinite. In MIMESIS, a minimal set of shapes is embedded to allow a basic visualization of the movement. But others particular algorithms have been used in the laboratory and sometimes even created by the laboratory itself.

6.1 A Flexible Coating Shape System

The Coating Shape system embedded within MIMESIS allows designing shapes for coating the generated movement, in the Projection Windows (Fig 8). Two coating streams are available.
The first, the "control view", offers a diagrammatic view of the model, as in the 3D conception Window.
The second, we call shape system, allows defining shapes on the basis of 'moving points'. A moving point can either correspond directly with a moving mass, or be defined from a set of moving masses by applying a

geometrical relation (geometric centre, intersection point, etc.).
On the basis of the moving points, the shape system will compute and display geometric shapes though OpenGL. A growing set of graphic primitives is available: geometric primitives (spheres, squares, circles, points, lines…), facets, polygons, computational geometry primitives (quadratics, nurbs…), implicit surfaces, Voronoï diagrams, Delaunay triangulation, etc. The coating shape system, when used, allows generating series of static images, or directly TGA movies.

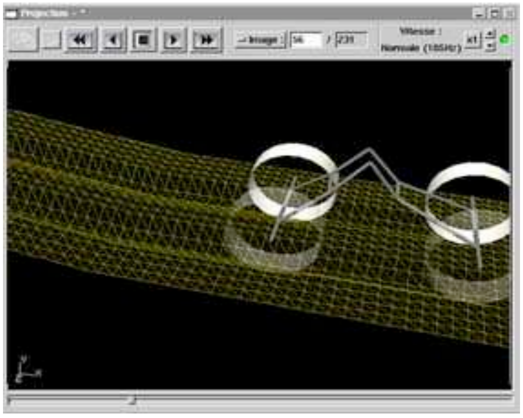

Figure 8: the projection window, which eventually allows controlling the embedded threaded simulator

6.2 Third-Party Visualization Algorithms

In addition to the embedded Coating Shape System, the MIMESIS framework offers a set of original physics-based dynamic coating techniques designed in the laboratory and implemented as separated programs [17].

6.3 Exportation of GMS files

Finally, MIMESIS allows exporting *gesture* (or *movement*) files in various formats. The rendering of the movement can consequently be performed with any 3D rendering technique, by importing the produced files in Computer Graphics software (as MAYA, 3DS max, Blender, After Effect, etc.) as evolution functions.

7 Examples

For years, a network of users has been set up in Europe joining together art schools, scientific vulgarization centers, and individuals. Additionally, various art students experiment with MIMESIS each year.
The models available in the library testify of the diversity of the category of movement (and animated images through the Coating Shape system and the third party physics-based coating algorithms) that can be obtained: mechanical structures, pendulums, vehicles, avalanches, flows, smokes, pasts and gels, dance movements, etc. We propose in the article a very small set of example (figures 9, 10, 11, 12, 13, 14).

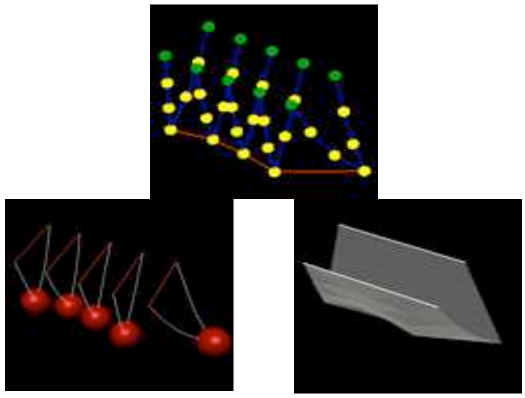

Figure 9: A simple Pendulum model. Up: diagrammatic view. Down: 2 coating using the coating shape system

Figure 10: waves and smoke rendered through the dynamic quoting algorithm.

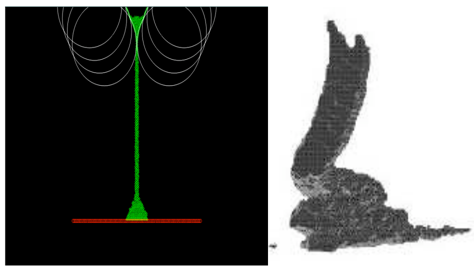

Figure 11: A 2D model of a past, coated with the embedded Coating Shape System, then with a third party coating algorithm

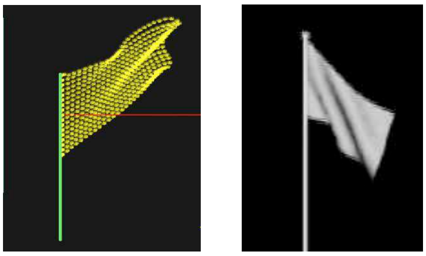

Figure 12: a flag. Left: diagrammatic view. Right: coated with Povray

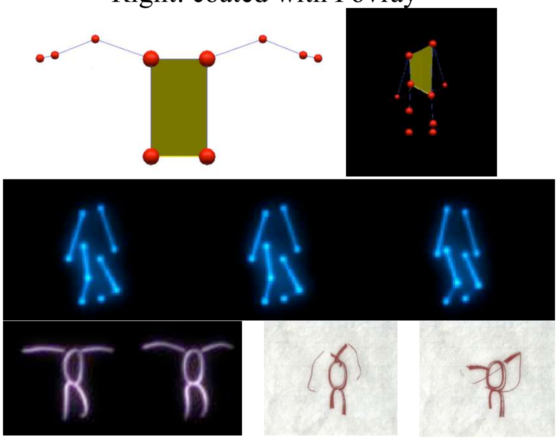

Figure 13: Top: Snapshots of MIMESIS models for synthesizing dance movements. Middle: MIMESIS coating. Bottom: Maya Coating [18]

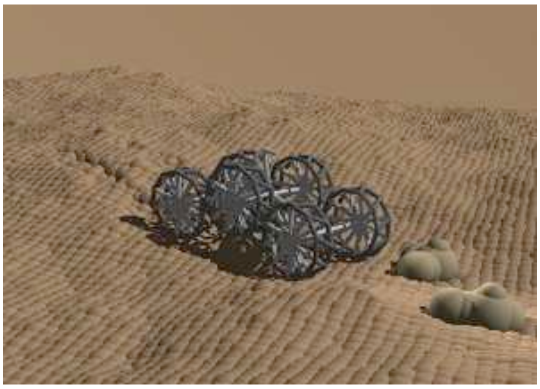

Figure 14: a model of a vehicle on a loose soil

8 Conclusions

The article presented MIMESIS, a design tool allowing the users to design interactively mass-interaction models and motion synthesis by means of such basic method. Numerous improvements are foreseen. Examples are: extending the language for allowing language-based interaction with the parameters and initial conditions, adding more complex shapes to the Coating Shape System, incorporating the original satellite coating techniques within the MIMESIS modeler itself, optimizing the software for a more fluid handling of huge models with hundred thousands physical modules, dynamical parameter modification, 3D-1D cooperation.

As a dedicated designing tool for mass-interaction simulation motor, focusing on motion modeling, MIMESIS is a complementary tool of 3D modeling systems or animation systems based on kinematics or Key-frame methods. By means of motion data import and export, it may participate to heterogeneous computer graphics modeling environments. Upstream and downstream cooperation with software as Povray, MAYA, 3DS max, Blender, or After Effect have been experimented by MIMESIS users. It has been successfully used to interactively produce complex motions by non-scientific users: water propagation, turbulences in smokes, crowd trajectories, vehicles evolving on granular sand, sand pilings, etc.). These results encourage us in our vision that to dispose of a design tool makes the practice of physically-based models significantly easier. Consequently, it is really a help to overcome the intrinsic difficulty of physical modeling by non-physicists. MIMESIS has been successfully installed and used in several schools of Arts and Arts Studios. It has supported several students' art works and newly used in three European artworks.

9 Acknowledgements

This work was supported by the FP6 Network of Excellence IST-2002-002114 - Enactive Interfaces, and by the French Ministry of Culture.

10 References

[1] G.S.P Miller. The motion dynamics of snakes and worms. *Computer Graphics*, vol. 22, SIGGRAPH'88, 169-178, Aug 1988.

[2] G.S.P. Miller, A. Pearce. Globular dynamics : a connected particle system for animating viscous fluids. *Computers & Graphics*, vol. 23(3):169-178, Elsevier, 1989.

[3] A. Luciani, S. Jimenez, J.-L. Florens, C. Cadoz, O. Raoult. Computational physics: a modeler simulator for animated physical

objects. *Proceedings of the European Computer Graphics Conference and Exhibition*. Eurographics'91, pp 425-436, Vienna, Austria, Elsevier Ed, Sep. 1991.

[4] D. Tonnesen. Modeling liquids and solids using thermal particles. *Graphics Interface'91,* 255-262, 1991.

[5] D. Terzopoulos, J. Platt, K. Fleisher. Heating and melting deformable models : from goop to glop. *Graphics Interface'89,* 219-226, 1989.

[6] Sodaplay java applet - http://www.sodaplay.com

[7] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. *Proceedings of Graphics Interface 1995*, Graphic Interface'95, pp 147-154, Quebec City, Canada, (May 1995).

[8] A. Witkin, D. Baraff, and M. Kass. Physically based modeling. SIGGRAPH 2001 COURSE NOTES #25, August 2001.

[9] A. Nealen, M. Müller, R. Keiser, E. Boxerman, M. Carlson. Physically Based Deformable Models in Computer Graphics. *Eurographics 2005 STAR*, Dublin, Ireland, Aug 29 – Sept 02, 2005.

[10] Reeves W. T. Particle systems: a technique for modelling a class of fuzzy objects. P. P. Tanner, editor, *Proceedings of the 10th annual conference on computer graphics and interactive techniques*, volume 2:91-108, SIGGRAPH'83, Detroit, Michigan, 1983.

[11] D. Greenspan. *Discrete Models*. Applied Mathematics Collection. Addison-Wesley (1973).

[12] D. Greenspan. *Particle Modeling*. Birkhauser Ed (1997).

[13] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, P. Volino. Collision Detection for Deformable Objects. Computer Graphics Forum, vol.24(1), March 2005, 61-81.

[14] A. Luciani, M. Evrard, N. Castagné, D. Couroussé, J.-L. Florens, C. Cadoz. A basic gesture and motion format for virtual reality multisensory applications. *Proceedings of the GRAPP conference*, February 2006.

[15] D.A. Norman. The Design of Everyday Things. Doubleday, New York, 1990.

[16] Shneiderman B. Direct Manipulation: A Step Beyond Programming Languages. *IEEE*, vol. 16, no. 8, pp 57-69 (1983).

[17] A. Habibi, A. Luciani. Dynamic particle coating. *Transactions on visualization and computer graphics*, , vol. 8, pp.383-394 , 2002/10-2002/12.

[18] C. Hsieh, A. Luciani. Physically-based particle modeling for dance verbs. *Proceedings of CASA 2006.*